

# BOUT++ overview

Ben Dudson and the BOUT++ team

BOUT++ workshop, LLNL  
14<sup>th</sup> August 2018



This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

# Acknowledgements



Many people have contributed to BOUT++

30 contributors to the github version

17 since version 3.0 (Oct 2016), with 3813 commits (~ 5.8 per day)

Particular thanks to:

Peter Hill, David Dickinson, Joseph Parker, David Schworer, John Omotani, Michael Loiten, Jens Madsen, Jarrod Leddy, Nick Walkden, Adam Dempsey, Haruki Seto, Brendan Shanahan, Erik Grinaker, Xiang Liu, Maxim Umansky, Matt Thomas



**GitHub**



**Travis CI**



**docker**



**slack**



**Read the Docs**

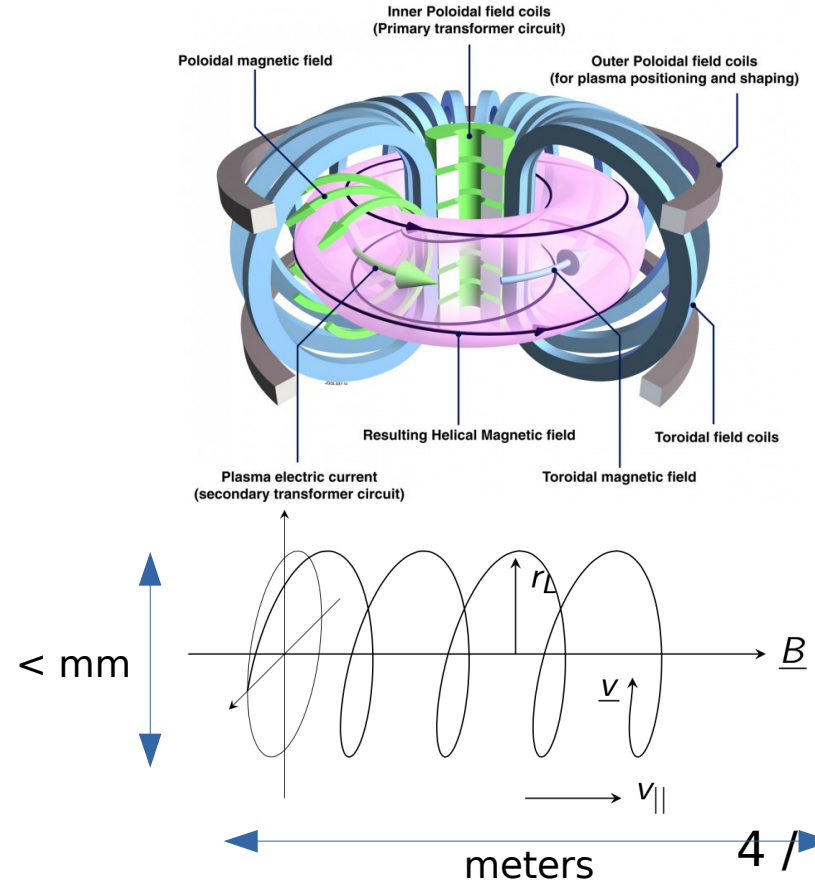
# Outline

- Plasma modelling, drift-reduced models
- BOUT++ code structure
- Recent changes (last ~ 2-3 years)
  - Electromagnetic field solvers, global modes
  - Complex magnetic geometries (FCI scheme)
- Getting started
- Contributing to BOUT++

# Tokamak plasmas

## Typical parameters

- Pressure  $\sim 1$  atmosphere
- Temperature  $\sim 100$  million  $^{\circ}\text{C}$
- Density  $\sim 10^{20} \text{ m}^{-3}$   
( $3.3 \times 10^{-7} \text{ kg/m}^3$  . Air  $\sim 1.2 \text{ kg/m}^3$ )
- Sound speed  $\sim 10^5 - 10^6 \text{ m/s}$
- Mean free paths  $\sim 1 - 100 \text{ m}$
- Ion gyrofrequency  $\sim \text{GHz}$
- Global evolution timescale  $\sim 10\text{s of ms}$



# Example model

A simple model can be derived from continuity of particles and charge:

$$\frac{\partial n}{\partial t} + \nabla \cdot (\underline{v}n) = 0$$

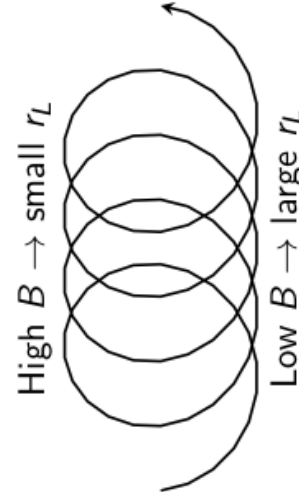
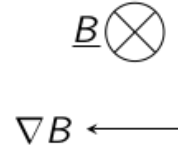
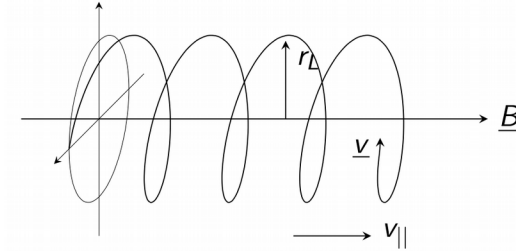
$$\nabla \cdot \underline{J} = 0$$

Particles drift due to electric fields, magnetic inhomogeneities and time-varying fields

$$\underline{v} = \underline{bv}_{||} + \frac{\underline{E} \times \underline{B}}{B^2} + \underbrace{\frac{T}{B} \frac{\underline{B} \times \nabla B}{B^2}}_{\text{Ignore for ions } (T_i = 0)} + \underbrace{\frac{m}{qB^2} \frac{d\underline{E}_{\perp}}{dt}}_{\text{Ignore for electrons } (m_e \text{ small})}$$

Ignore for ions ( $T_i = 0$ )

Ignore for electrons ( $m_e$  small)



# Example model

Substituting this velocity into density and current equations gives:

$$\frac{\partial n}{\partial t} = \underbrace{-\nabla \cdot (n \underline{b} v_{||e})}_{\sim 2/R_c} - \nabla \cdot \left( n \frac{\underline{E} \times \underline{B}}{B^2} \right) - \nabla \cdot \left( \frac{nT}{B} \underbrace{\frac{\underline{B} \times \nabla B}{B^2}}_{\sim 2/R_c} \right)$$

$$\nabla \cdot \left( \frac{m_i n}{e B^2} \frac{d \underline{E}_\perp}{dt} \right) = -\nabla \cdot J_{||} - \nabla \cdot \left( \frac{-enT}{B} \frac{\underline{B} \times \nabla B}{B^2} \right)$$

Assuming incompressible, neglecting ion parallel flow, and making thin layer (Boussinesq) approximation gives:

$$\frac{\partial n}{\partial t} = -\frac{1}{B} \underline{b} \times \nabla \phi \cdot \nabla n + \frac{1}{e} \nabla \cdot J_{||} + \frac{2T_e}{BR_c} \frac{\partial n}{\partial z}$$

$$\frac{\partial \omega}{\partial t} = -\frac{1}{B} \underline{b} \times \nabla \phi \cdot \nabla \omega + \nabla \cdot J_{||} + \frac{2eT_e}{BR_c} \frac{\partial n}{\partial z}$$

$$\omega = \frac{m_i n_0}{B^2} \nabla_\perp^2 \phi$$

# Example model

$$\frac{\partial n}{\partial t} = -\frac{1}{B} \underline{b} \times \nabla \phi \cdot \nabla n + \frac{1}{e} \nabla \cdot J_{\parallel} + \frac{2T_e}{BR_c} \frac{\partial n}{\partial z}$$

$$\frac{\partial \omega}{\partial t} = \boxed{-\frac{1}{B} \underline{b} \times \nabla \phi \cdot \nabla \omega} + \nabla \cdot J_{\parallel} + \frac{2eT_e}{BR_c} \frac{\partial n}{\partial z}$$

Incompressible fluid e.g.  
Arakawa '65 and extensions

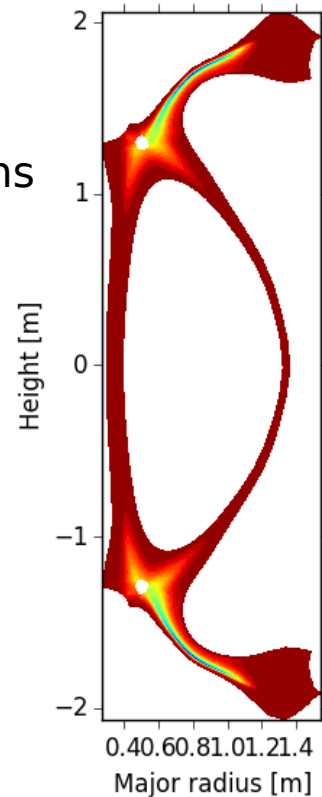
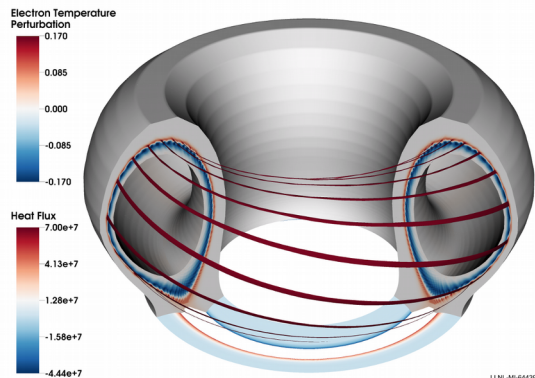
$$\omega = \frac{m_i n_0}{B^2} \nabla_{\perp}^2 \phi$$

Gradients along  
magnetic field  
(compressible fluid)

Need to invert Laplacian to evaluate electric potential.  
→ Spectral methods, multigrid schemes

# BOUT++

- An open-source library for solving PDEs in curvilinear geometry
- Specialised operators for plasma applications
- Finite difference / volume code in 3D mapped multi-block geometry
- Solves nonlinearly coupled hyperbolic, parabolic and elliptic equations
- MPI parallelised, scales to  $\sim 4,000$  cores (depending on problem)
- Turbulence  $\sim 10^6$ - $10^8$  unknowns,  $\sim 10^5$  core-hours





# BOUT++: example model



<https://github.com/boutproject/BOUT-dev/tree/master/examples/blob2d>

$$\frac{\partial n}{\partial t} = -\frac{1}{B} \underline{b} \times \nabla \phi \cdot \nabla n + \frac{1}{e} \nabla \cdot J_{\parallel} + \frac{2T_e}{BR_c} \frac{\partial n}{\partial z}$$

$$\frac{\partial \omega}{\partial t} = -\frac{1}{B} \underline{b} \times \nabla \phi \cdot \nabla \omega + \nabla \cdot J_{\parallel} + \frac{2eT_e}{BR_c} \frac{\partial n}{\partial z}$$

$$\omega = \frac{m_i n_0}{B^2} \nabla_{\perp}^2 \phi$$

```
ddt(n) = -vE_Grad(n,phi+phi0)
        + Div_par(Jpar)
        + 2*DDZ(n)*(rho_s/R_c);
```

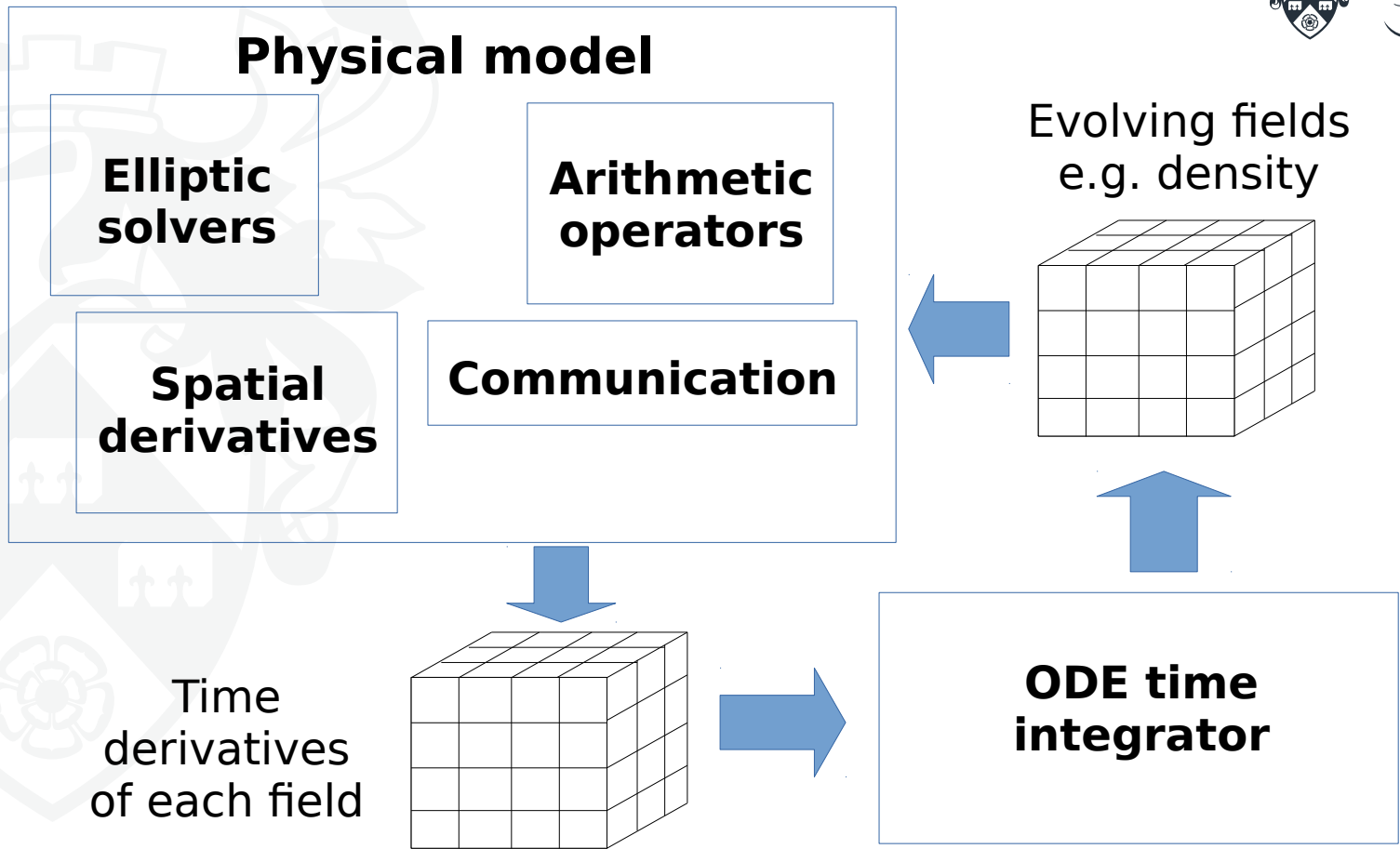
```
ddt(omega) = -bracket(phi,omega)
             + Div_par(Jpar)
             + 2*DDZ(n)*(rho_s/R_c);
```

```
phi = phiSolver->solve(omega);
```

# Outline

- Plasma modelling, drift-reduced models
- BOUT++ code structure
- Recent changes (last ~ 2-3 years)
  - Electromagnetic field solvers, global modes
  - Complex magnetic geometries (FCI scheme)
- Getting started
- Contributing to BOUT++

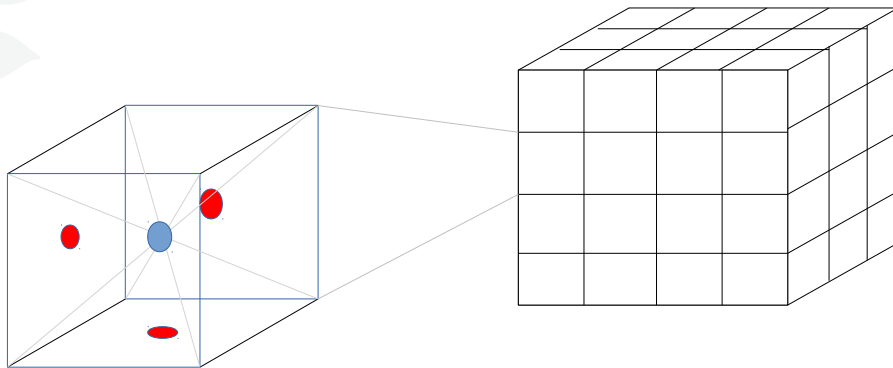
# BOUT++: Method of Lines



# BOUT++ mesh

3D mesh, locally logically Cartesian

Quantities usually **cell centre**; some staggered **cell face** values



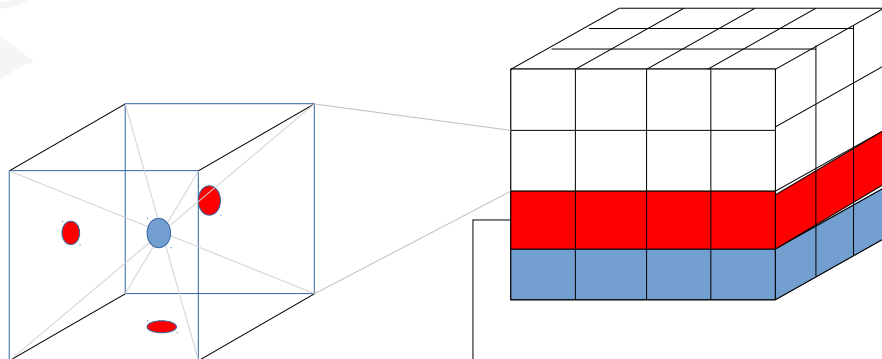
Represented as Field3D objects

```
Field3D n, T;  
...  
Field3D p = n * T;
```

# BOUT++ mesh

3D mesh, locally logically Cartesian

Guard/ghost cells used for boundaries and communication



Communication handled by Mesh:

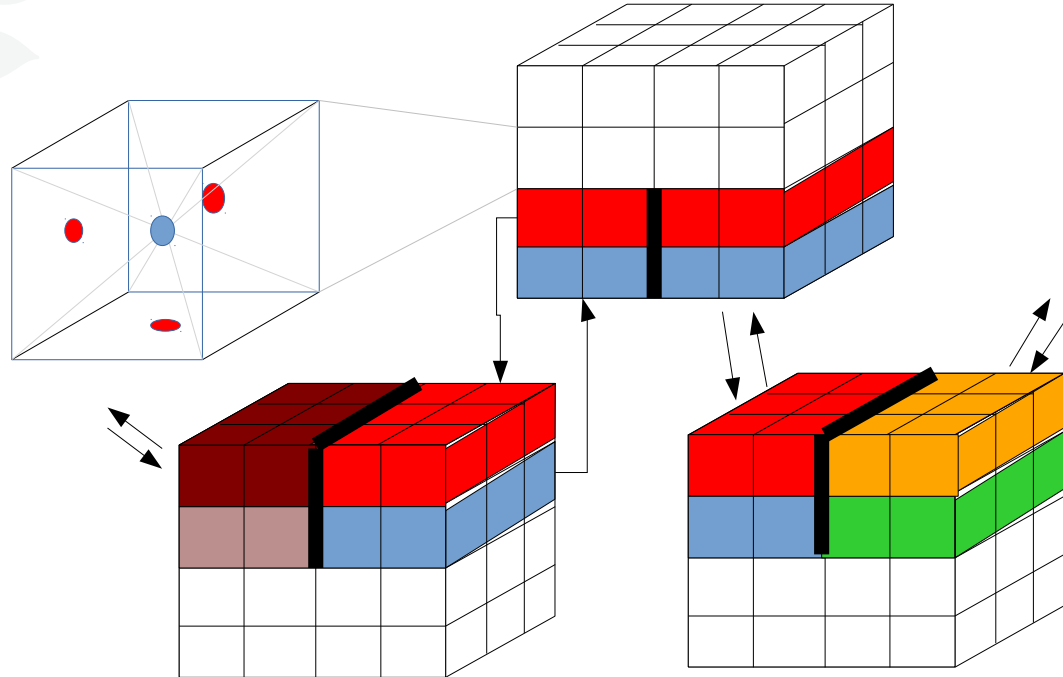
```
mesh->communicate(n, T);
```

Communication  
through guard cells

# BOUT++ mesh

3D mesh, locally logically Cartesian

More complicated topologies  
handled using branch cuts



# BOUT++ mesh

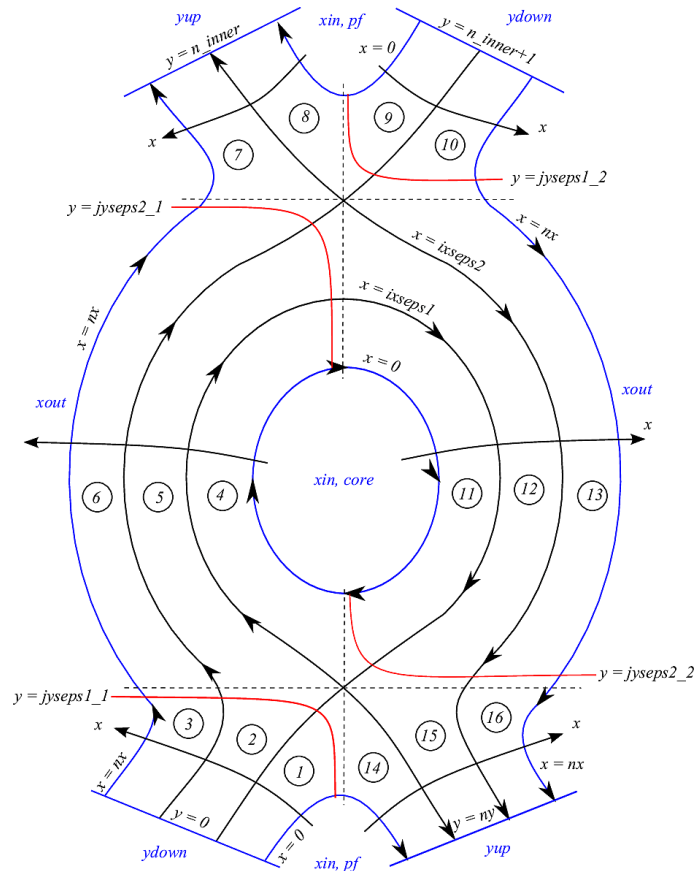
3D mesh, locally logically Cartesian

More complicated topologies handled using branch cuts

Tokamak edge simulations usually have at least one “X-point” and a hole in the domain.



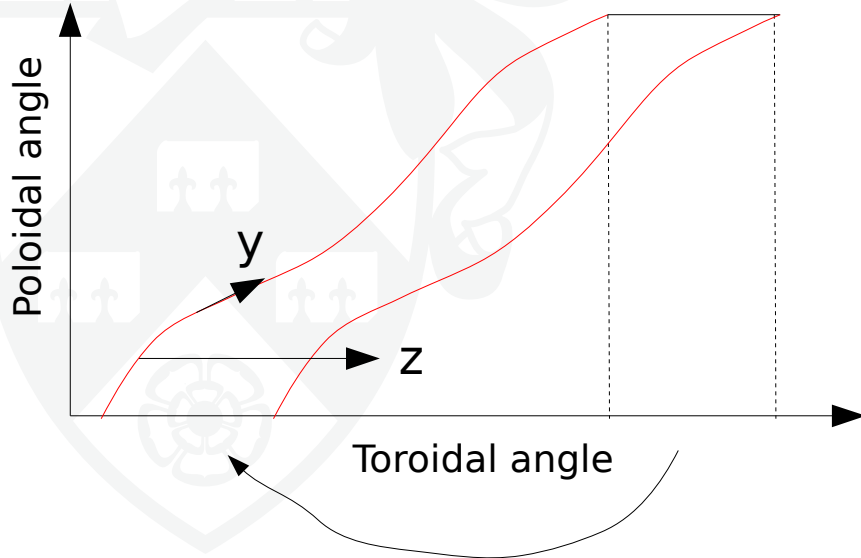
UNIVERSITY  
of York



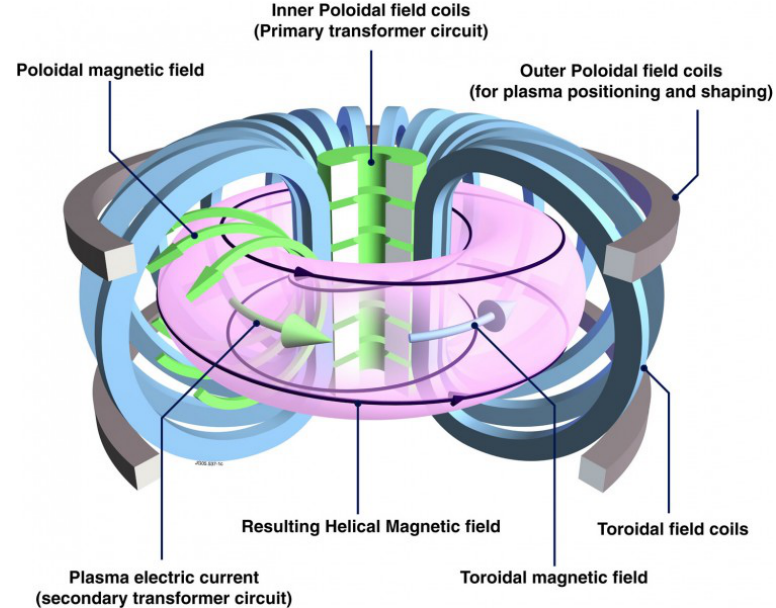
# Coordinates

Usually curved, usually non-orthogonal  
aligned to the helical magnetic field.

In the core of the plasma the domain is a  
twisted ribbon



Twist-shift boundary (matching) condition

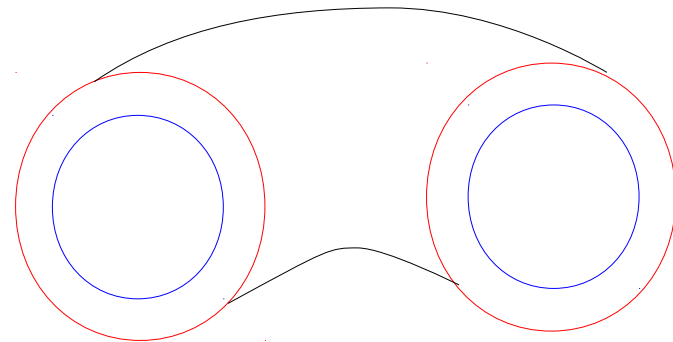
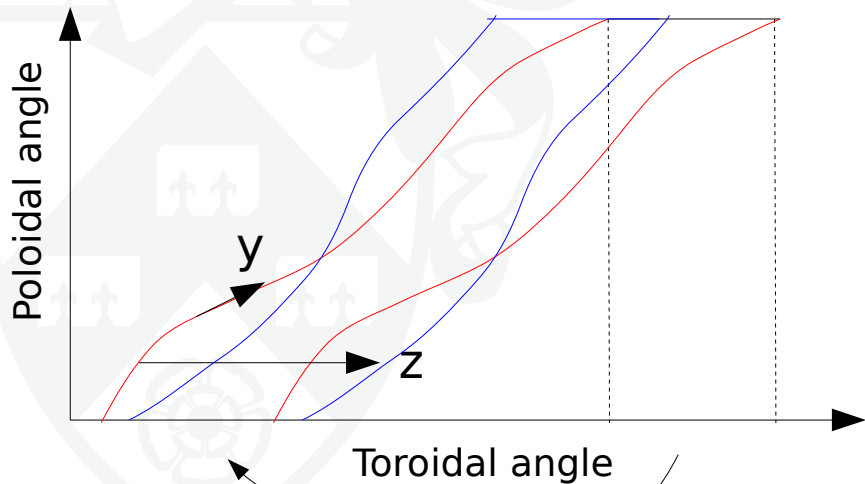




# Coordinates

Usually curved, usually non-orthogonal  
aligned to the helical magnetic field.

In the core of the plasma the domain is a  
twisted ribbon



Pitch of the magnetic field varies

Leads to shearing of coordinate system

Often necessary to re-map onto local  
coordinates to reduce shearing

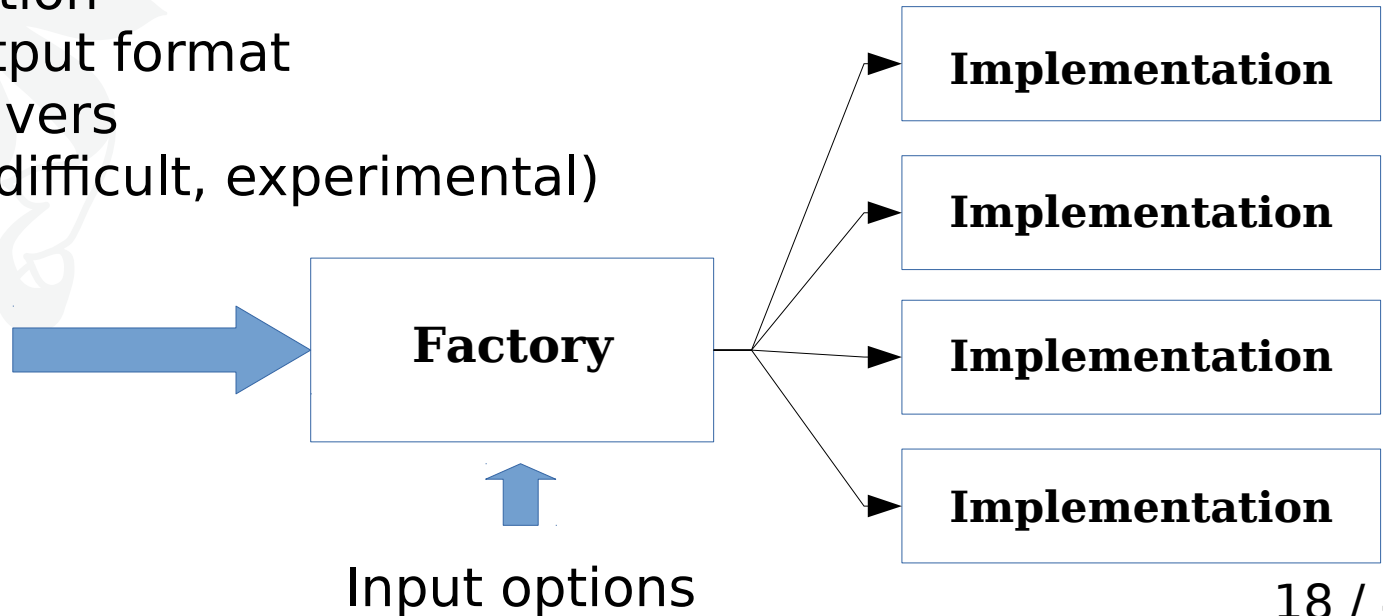
Shifts in toroidal angle  $\rightarrow$  FFTs

Twist-shift boundary (matching) condition

# Modular structure

Many parts of BOUT++ have multiple implementations.  
Changed with an input option, with no/minimal user code changes.

- Time integration
- File input/output format
- Laplacian solvers
- Mesh (more difficult, experimental)



# Outline

- Plasma modelling, drift-reduced models
- BOUT++ code structure
- Recent changes (last ~ 2-3 years)
  - Electromagnetic field solvers, global modes
  - Complex magnetic geometries (FCI scheme)
- Getting started
- Contributing to BOUT++

# Solving potentials

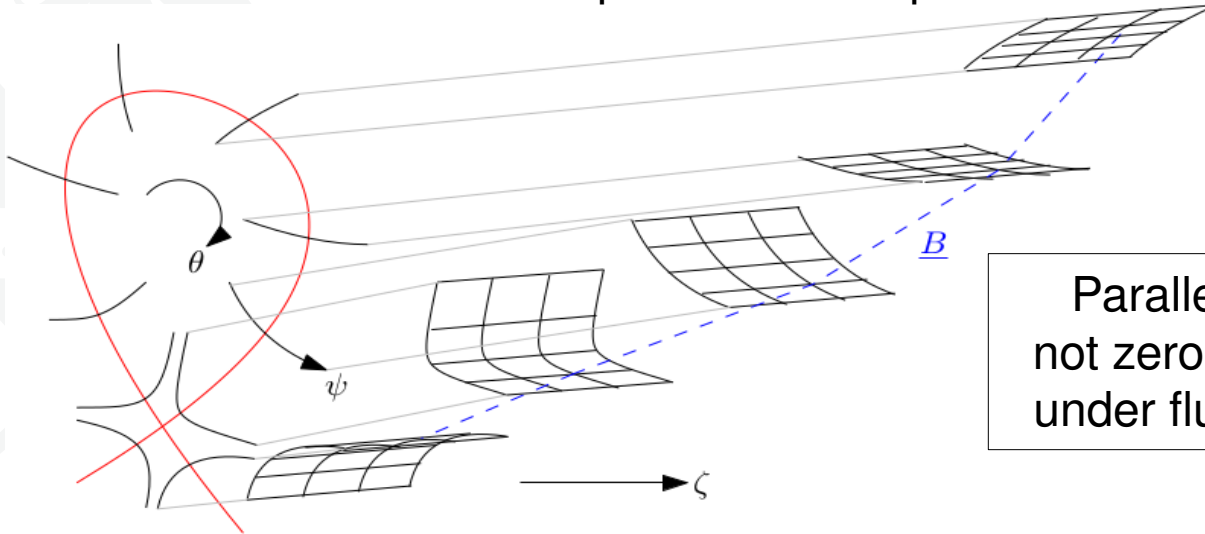
Need to invert a Laplacian to obtain potential from vorticity:

$$\nabla \cdot \left( \frac{m_i n}{B^2} \nabla_{\perp} \phi \right) = \frac{1}{J} \frac{\partial}{\partial u^i} \left( J \frac{m_i n}{B^2} g^{ij} (\nabla_{\perp} \phi)_j \right)$$

Coordinates

$$\begin{aligned} x &= \psi & y &= \theta \\ z &= \zeta - \int_{\theta_0}^{\theta} \frac{B_{\zeta} h_{\theta}}{B_{\theta} R} d\theta \end{aligned}$$

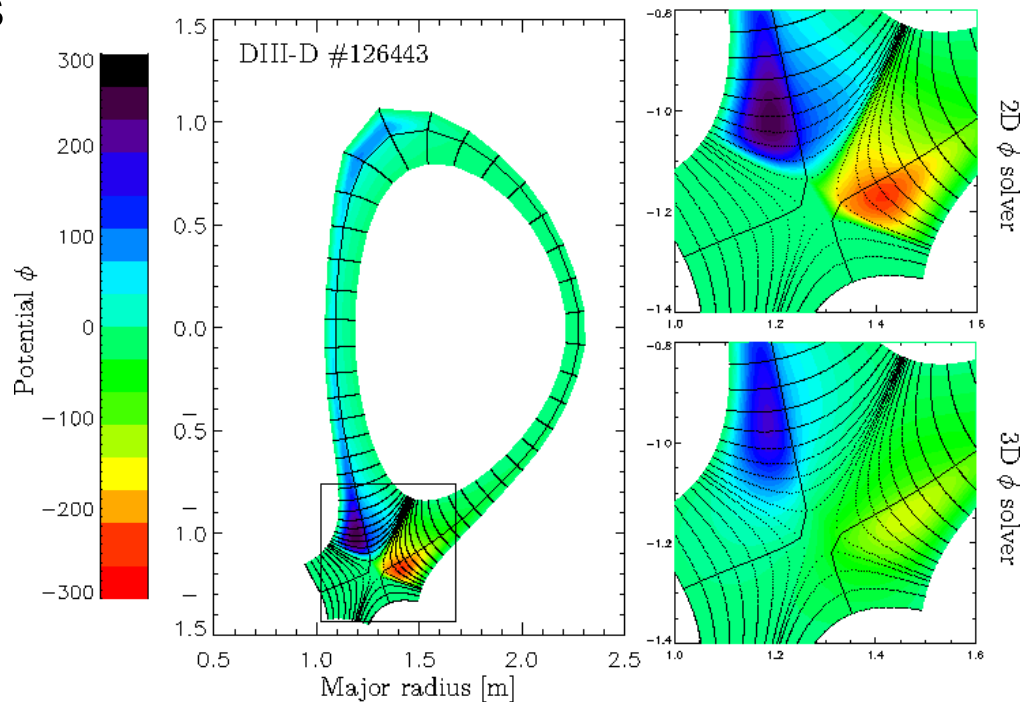
Field-aligned coordinates uses toroidal planes as drift planes:



Parallel component  
not zero, but neglected  
under flute assumption

# Solving $n=0$ modes

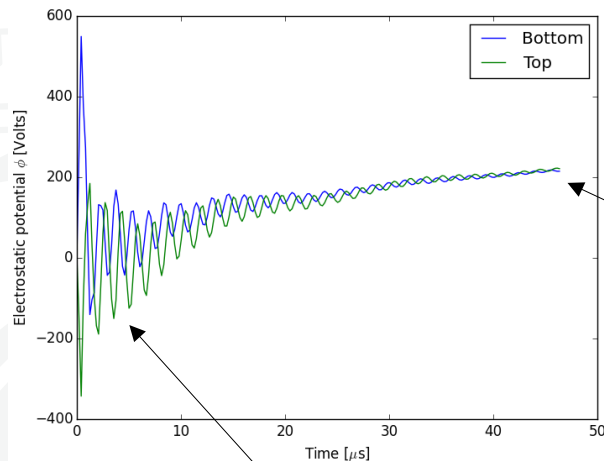
- Solving on toroidal planes results in large savings: Each X-Z slice can be solved separately
- For low  $n$  modes this simplification breaks down. Results in unphysical electric fields close to the X-point
- A 2D solver has been developed for the  $n=0$  mode using PETSc



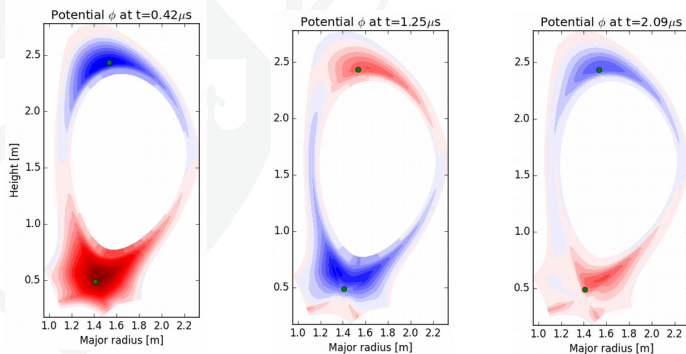
# Solving $n=0$ modes

## Drifts, currents

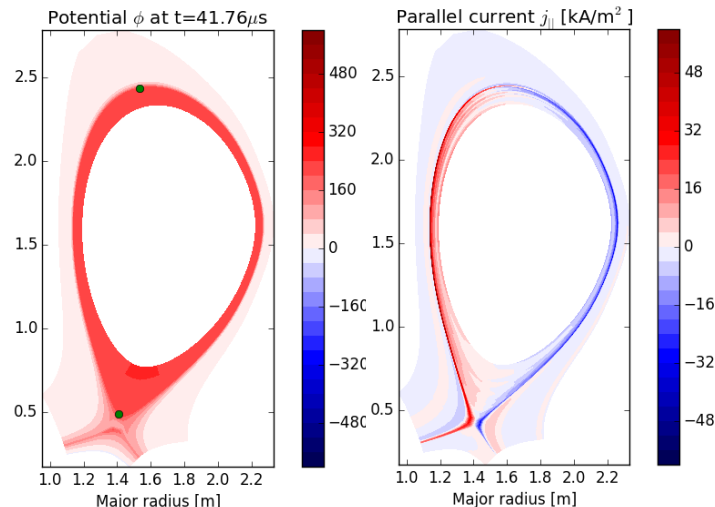
Evolve  $n=0$  electric field  
Transients and steady state



## Transient oscillation



## Steady state

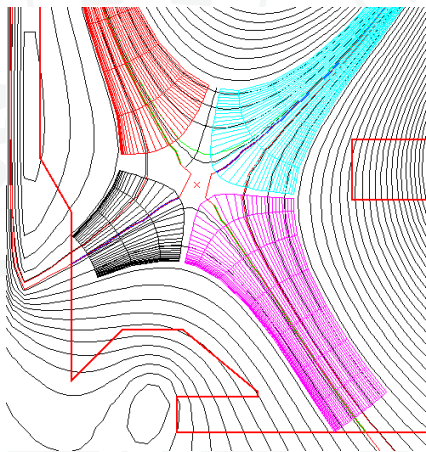


# Complicated magnetic fields

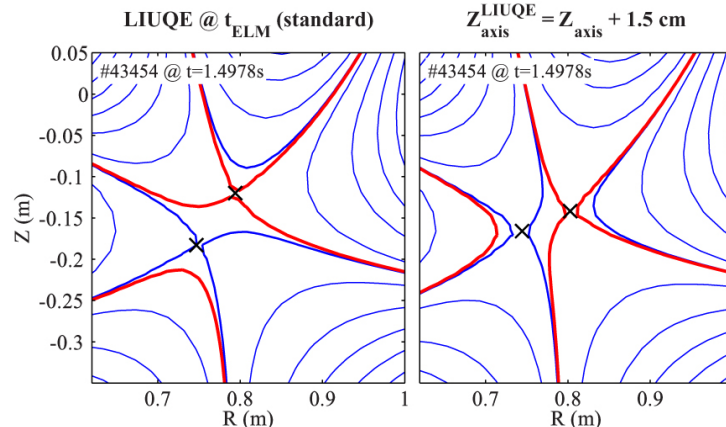


UNIVERSITY  
of York

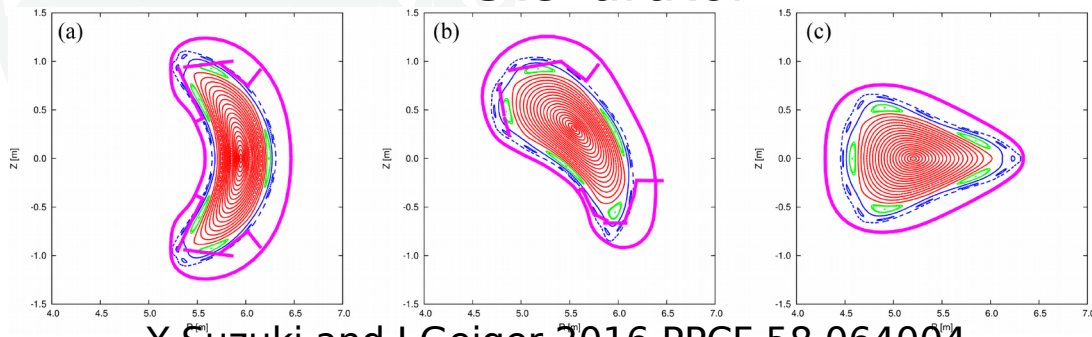
## X-point



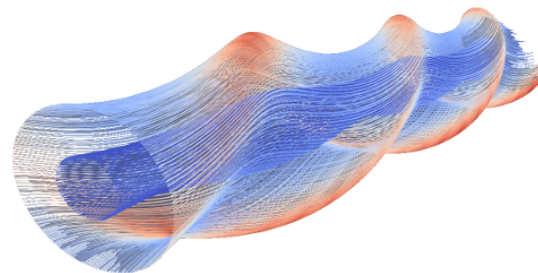
## Snowflake



## Stellarator



W.A.J. Vijvers et al 2014 Nucl. Fusion 54 023009



Y Suzuki and J Geiger 2016 PPCF 58 064004

# Flux coordinate independent



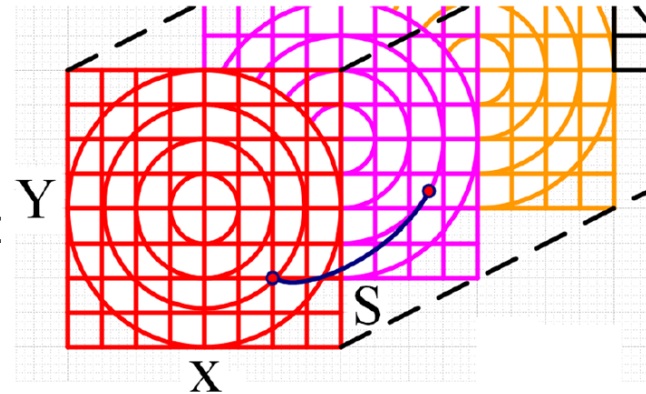
UNIVERSITY  
of York

## Field aligned grids are required in order to:

- Significantly reduce resolution in one dimension
- Accurately capture anisotropic transport

## Field-aligned coordinates have singularities

- Reduces convergence rate
- Can lead to numerical instability
- Difficult to resolve features close to the X-point



## The FCI approach is to use:

- Well-behaved local coordinates,
- field-line following and interpolation to calculate parallel derivatives

[1] F Hariri and M Ottaviani CPC **184** 2419 (2013)

[2] B Shanahan, P Hill and B Dudson. Journal of Physics; Conference Series **775**, 012012 (2016)

[3] A.Stegmeir et al. CPC 198, 139-153 (2016)

[4] P Hill, B Shanahan and B Dudson CPC **213**, 9-18 (2017)

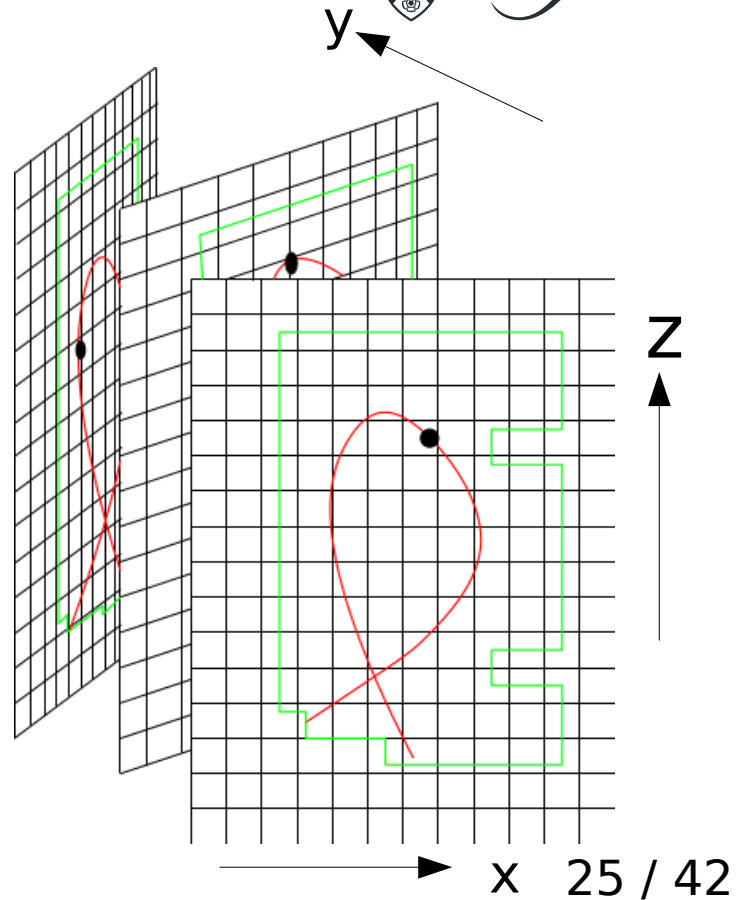


# Tokamak simulations

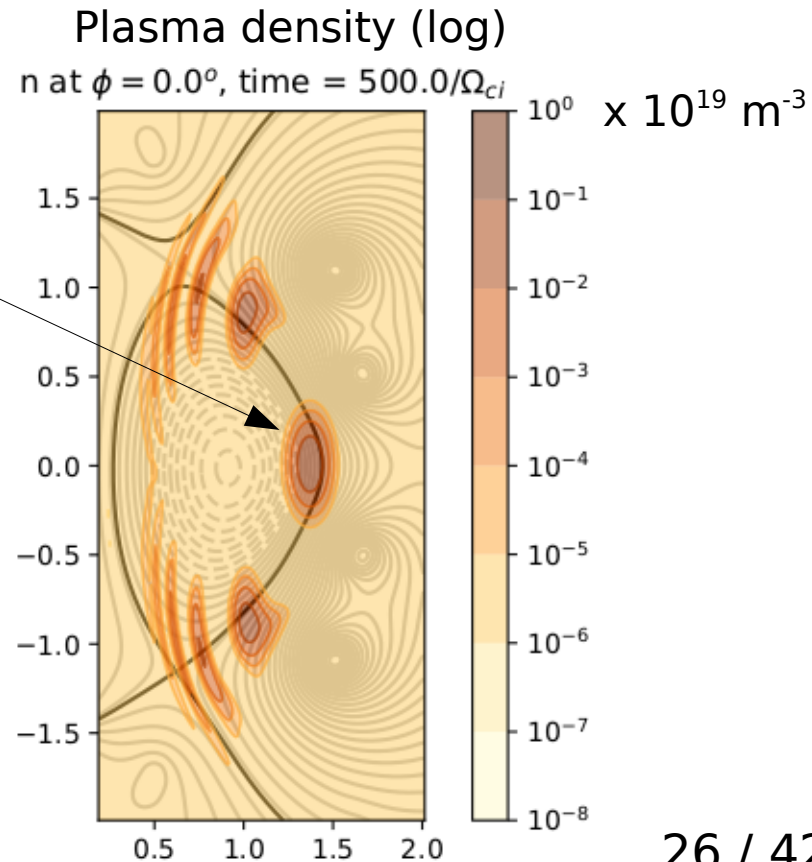
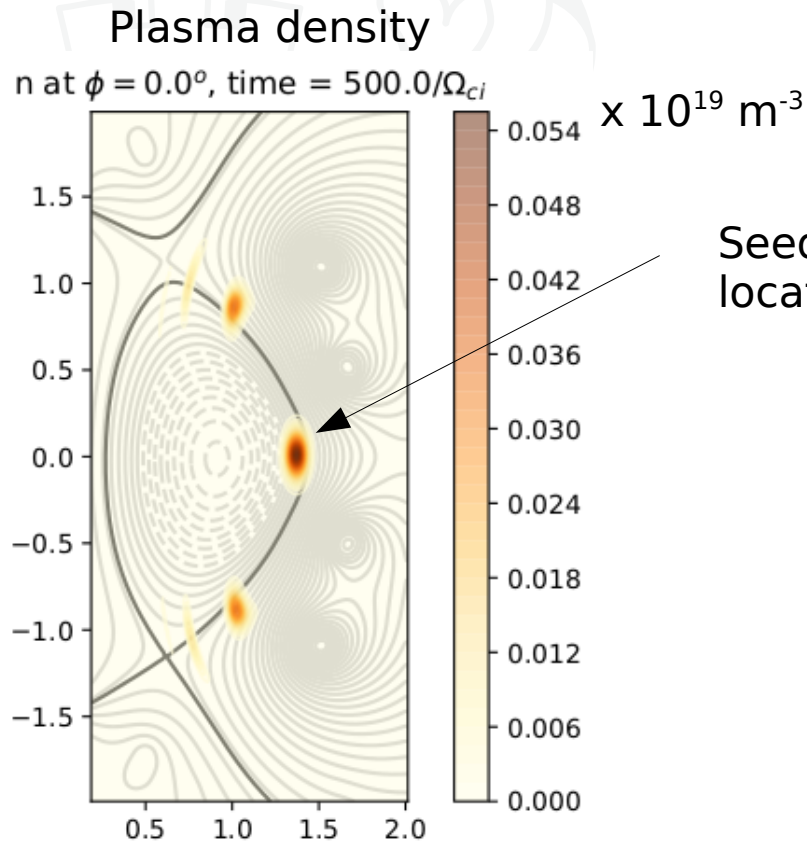
MAST geometry (shot 14220)  
 $R = 0.2 \rightarrow 2\text{m}$  ;  $Z = -1.5 \rightarrow +1.5\text{m}$

Simple sheath boundary conditions  
at walls:  $v_{\parallel} = cs$

Electromagnetic field evolved (Alfven  
waves), but B perturbation **not**  
included (for now)



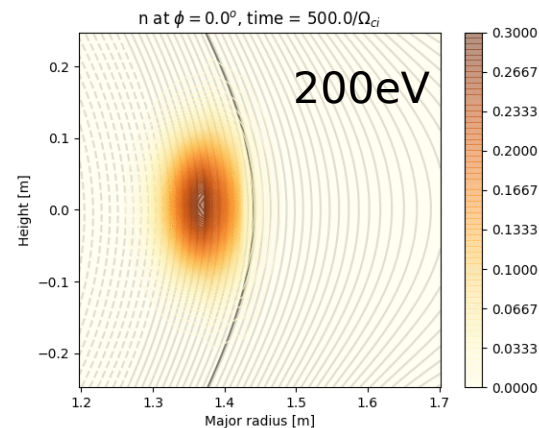
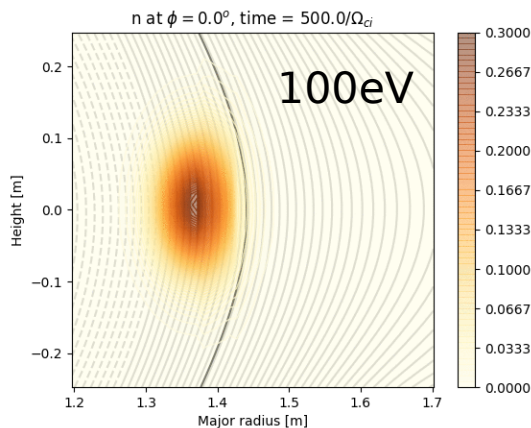
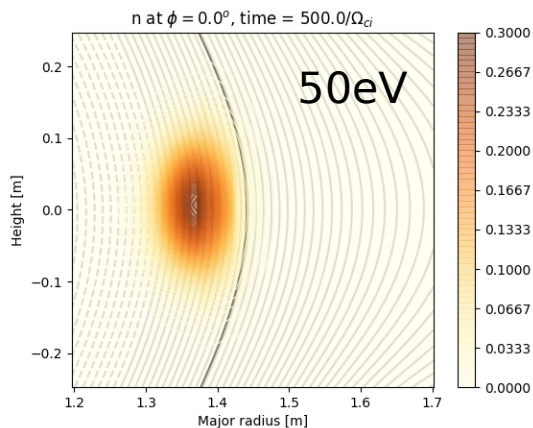
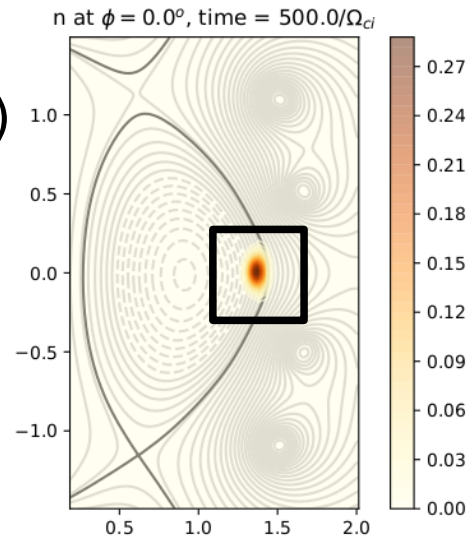
# Tokamak filaments (n=5)



# Tokamak filaments

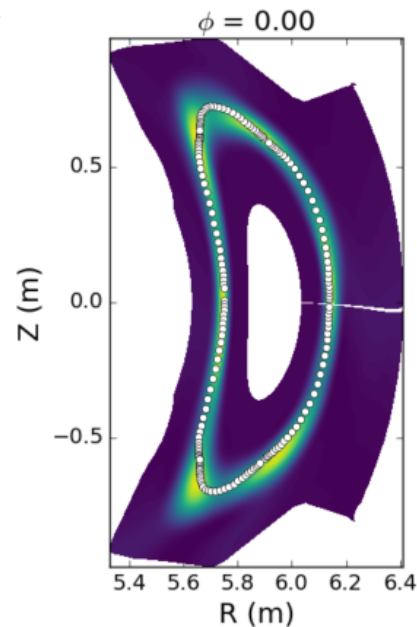
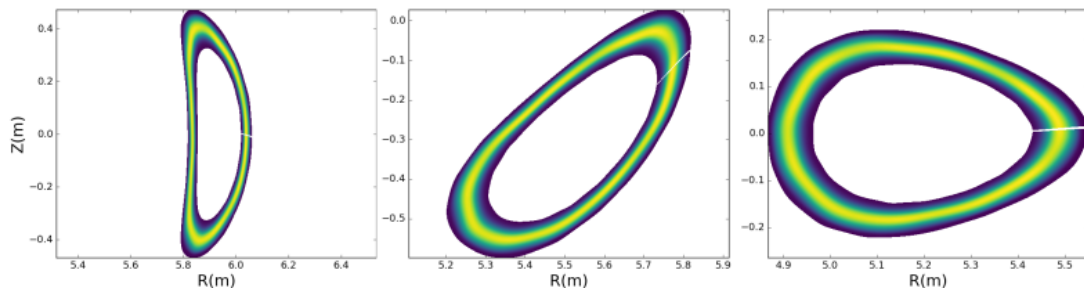
(Under-resolved proof-of-principle)

- Start with an isolated plasma “ball” (no background)
- Evolve only parallel diffusion ( $10^5$  m/s) for  $\sim 10\mu\text{s} \rightarrow$  Filament
- Turn on drifts, electric fields



# Stellarator simulations

- New grid generator for (nearly) arbitrary shapes
- Initial blob studies in “straight stellarators”
- Simulations of Wendelstein 7-X started



(b) Simulated surfaces for the  
bean-shaped-cross section

FIG. 14: Three cross sections of the Wendelstein 7-X stellarator indicating flux surfaces as traced by a parallel heat diffusion equation in BSTING

# Other recent changes

Some of the many fixes and new features

1. Ability to handle complex 3D magnetic field structures
2. Better electromagnetic field solvers
3. Running BOUT++ from Python
4. Faster, clearer loops, C++11 features\*\*
5. Better differential operators, flux-conserving schemes
6. More usable, easier to install, debug, optimise, extend

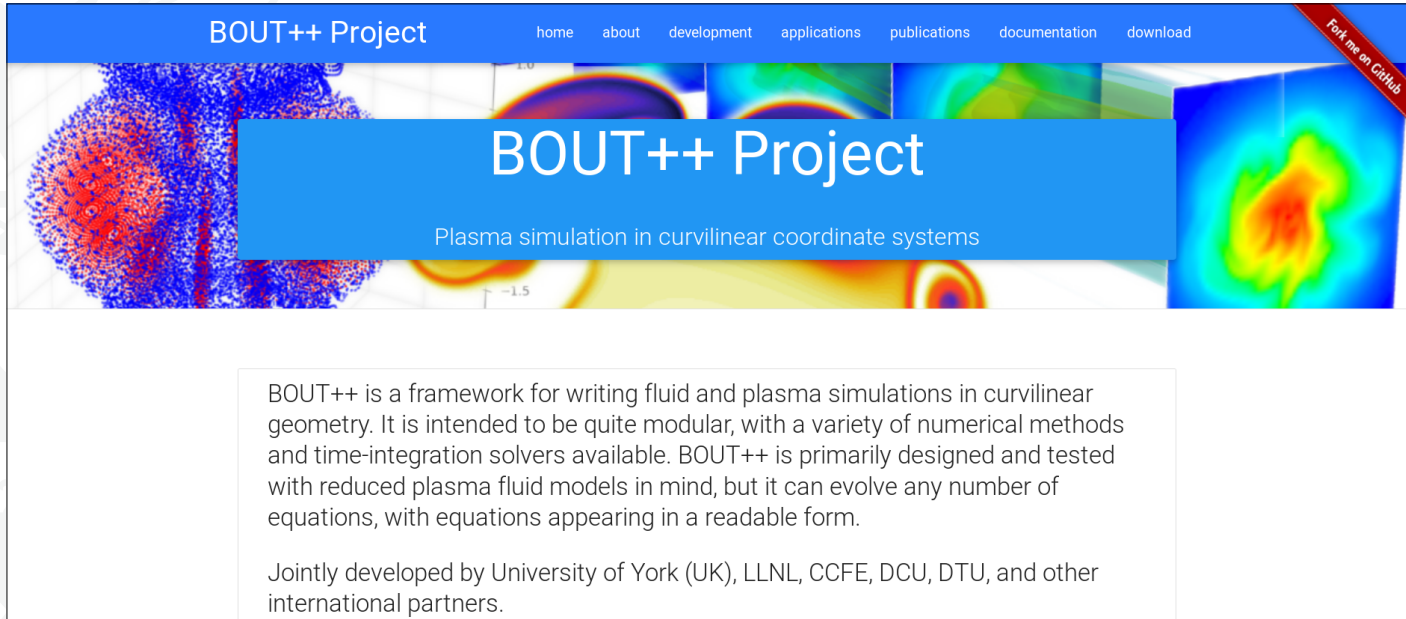
\*\* See performance talk (tomorrow)

# Outline

- Plasma modelling, drift-reduced models
- BOUT++ code structure
- Recent changes (last ~ 2-3 years)
  - Electromagnetic field solvers, global modes
  - Complex magnetic geometries (FCI scheme)
- Getting started
- Contributing to BOUT++

# Getting started

<http://boutproject.github.io/>



The screenshot shows the BOUT++ Project website. At the top is a blue navigation bar with the text 'BOUT++ Project' and links for 'home', 'about', 'development', 'applications', 'publications', 'documentation', and 'download'. Below the navigation bar is a large banner image featuring several colorful plasma simulation visualizations. Overlaid on the banner is a blue box with the text 'BOUT++ Project' and 'Plasma simulation in curvilinear coordinate systems'. To the right of the banner is a red ribbon with the text 'Fork me on GitHub'. Below the banner is a white box containing text about the project.

BOUT++ Project

home about development applications publications documentation download

**BOUT++ Project**

Plasma simulation in curvilinear coordinate systems

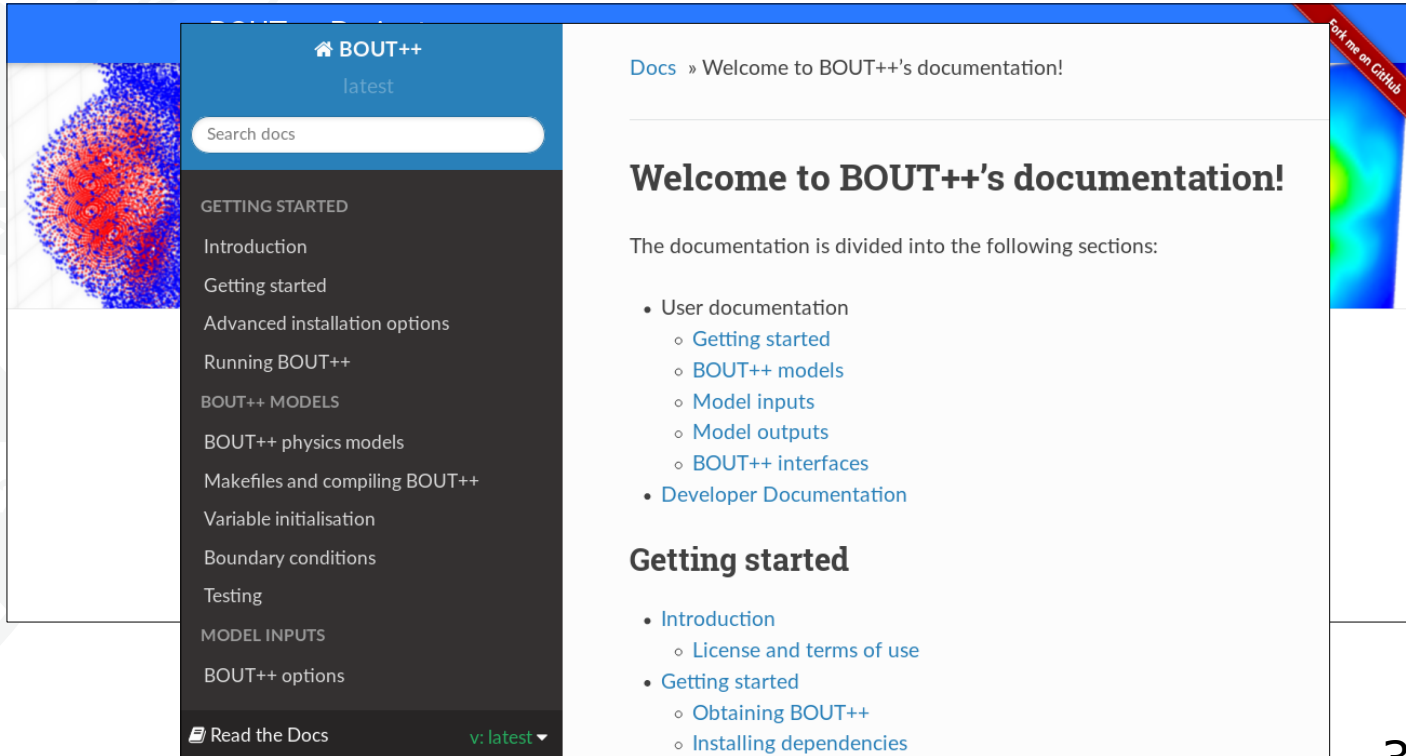
Fork me on GitHub

BOUT++ is a framework for writing fluid and plasma simulations in curvilinear geometry. It is intended to be quite modular, with a variety of numerical methods and time-integration solvers available. BOUT++ is primarily designed and tested with reduced plasma fluid models in mind, but it can evolve any number of equations, with equations appearing in a readable form.

Jointly developed by University of York (UK), LLNL, CCFE, DCU, DTU, and other international partners.

# Getting started

<http://bout-dev.readthedocs.io>



The screenshot shows the BOUT++ documentation website. The left sidebar contains a navigation menu with the following sections: GETTING STARTED (Introduction, Getting started, Advanced installation options, Running BOUT++), BOUT++ MODELS (BOUT++ physics models, Makefiles and compiling BOUT++, Variable initialisation, Boundary conditions, Testing), and MODEL INPUTS (BOUT++ options). The main content area displays the title 'Welcome to BOUT++'s documentation!' and a list of sections: User documentation (Getting started, BOUT++ models, Model inputs, Model outputs, BOUT++ interfaces) and Developer Documentation. Below this, the 'Getting started' section is expanded, showing Introduction (License and terms of use) and Getting started (Obtaining BOUT++, Installing dependencies). A red banner in the top right corner reads 'Work me on GitHub'.

**BOUT++**  
latest

Search docs

GETTING STARTED

- Introduction
- Getting started
- Advanced installation options
- Running BOUT++

BOUT++ MODELS

- BOUT++ physics models
- Makefiles and compiling BOUT++
- Variable initialisation
- Boundary conditions
- Testing

MODEL INPUTS

- BOUT++ options

Read the Docs v: latest

Docs » Welcome to BOUT++'s documentation!

## Welcome to BOUT++'s documentation!

The documentation is divided into the following sections:

- User documentation
  - [Getting started](#)
  - [BOUT++ models](#)
  - [Model inputs](#)
  - [Model outputs](#)
  - [BOUT++ interfaces](#)
- Developer Documentation

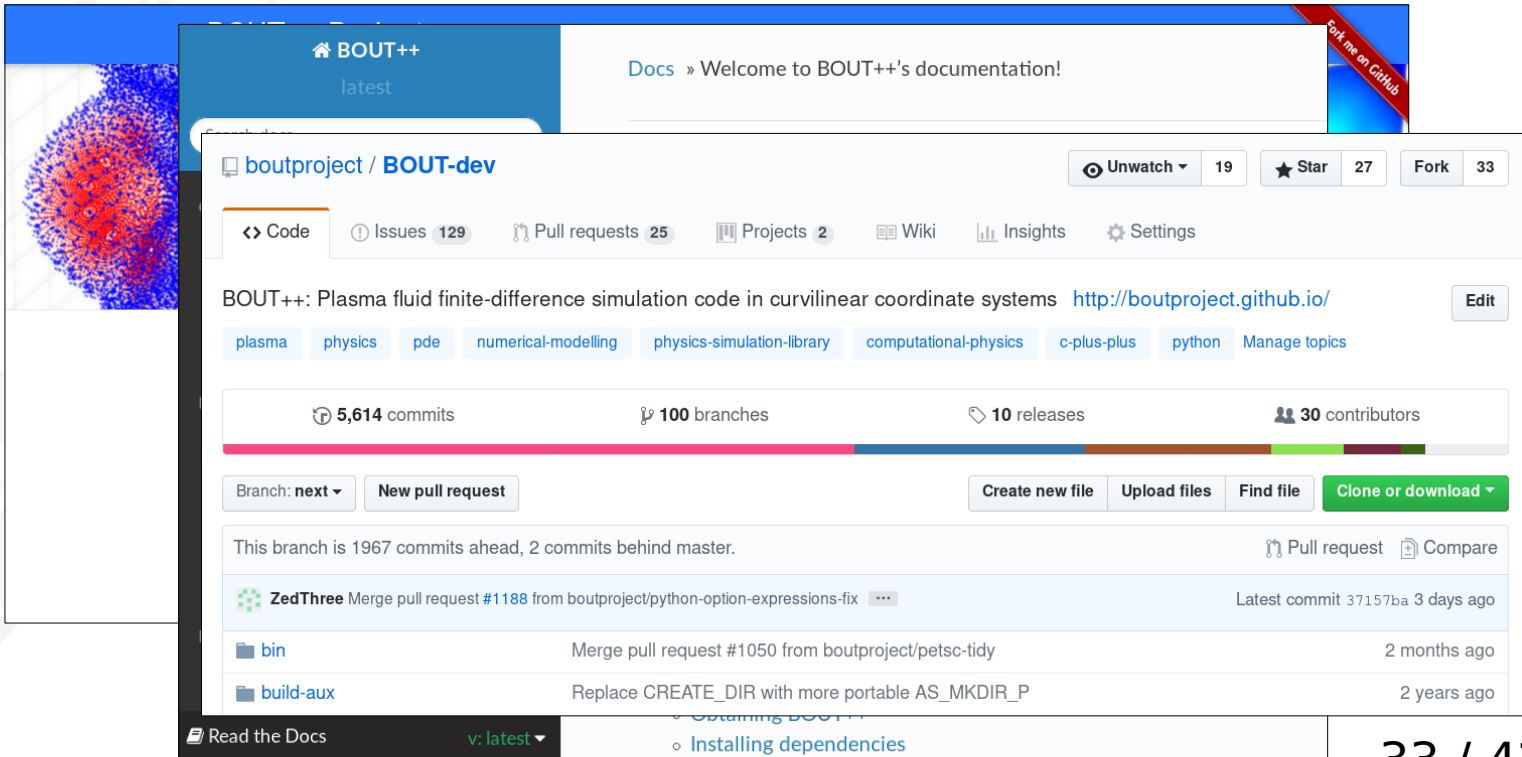
## Getting started

- Introduction
  - [License and terms of use](#)
- Getting started
  - [Obtaining BOUT++](#)
  - [Installing dependencies](#)



# Getting started

<https://github.com/boutproject/BOUT-dev>



The screenshot shows the GitHub repository page for **boutproject / BOUT-dev**. The repository is described as "Plasma fluid finite-difference simulation code in curvilinear coordinate systems" with a link to <http://boutproject.github.io/>. It has 5,614 commits, 100 branches, 10 releases, and 30 contributors. The current branch is **next**, which is 1967 commits ahead and 2 commits behind master. Recent pull requests are listed, including a merge from `boutproject/python-option-expressions-fix` and another from `boutproject/petsc-tidy`. The repository is tagged as "latest" and includes a "Read the Docs" link.

**BOUT++**  
latest

Docs » Welcome to BOUT++'s documentation!

boutproject / **BOUT-dev** Unwatch 19 Star 27 Fork 33

<> Code Issues 129 Pull requests 25 Projects 2 Wiki Insights Settings

BOUT++: Plasma fluid finite-difference simulation code in curvilinear coordinate systems <http://boutproject.github.io/> Edit

plasma physics pde numerical-modelling physics-simulation-library computational-physics c-plus-plus python Manage topics

5,614 commits 100 branches 10 releases 30 contributors

Branch: next New pull request Create new file Upload files Find file Clone or download

This branch is 1967 commits ahead, 2 commits behind master. Pull request Compare

**ZedThree** Merge pull request #1188 from boutproject/python-option-expressions-fix Latest commit 37157ba 3 days ago

bin Merge pull request #1050 from boutproject/petsc-tidy 2 months ago

build-aux Replace CREATE\_DIR with more portable AS\_MKDIR\_P 2 years ago

Read the Docs v: latest Installing dependencies

# Install docker



Jarrold Leddy (Tech-X)

- Provides a reproducible environment (like a virtual machine or BSD jail)

Ubuntu linux

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

Mac:

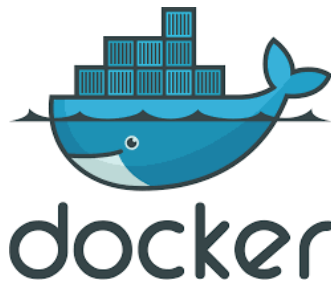
<https://docs.docker.com/docker-for-mac/install/>

Windows:

<https://docs.docker.com/docker-for-windows/install/>

Others:

<https://docs.docker.com/v17.12/install/>

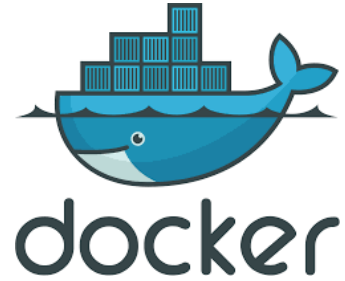


# BOUT++ docker image

Jarrold Leddy (Tech-X)



- Provides a reproducible environment (like a virtual machine or BSD jail)
- Preinstalled with PETSc, SLEPC, SUNDIALS, BOUT++, editors and python analysis tools



<https://hub.docker.com/r/boutproject/bout-next/>

```
$ sudo docker pull boutproject/boutproject/bout-next:9f4c663-petsc
```

```
$ sudo docker run --rm -it boutproject/bout-next:9f4c663-petsc
```

# Outline

- Plasma modelling, drift-reduced models
- BOUT++ code structure
- Recent changes (last ~ 2-3 years)
  - Electromagnetic field solvers, global modes
  - Complex magnetic geometries (FCI scheme)
- Getting started
- Contributing to BOUT++

# Development process



[boutproject](#) / **BOUT-dev** Unwatch 19 Star 27 Fork 33

[Code](#) [Issues 129](#) [Pull requests 25](#) [Projects 2](#) [Wiki](#) [Insights](#) [Settings](#)

BOUT++: Plasma fluid finite-difference simulation code in curvilinear coordinate systems <http://boutproject.github.io/> [Edit](#)

[plasma](#) [physics](#) [pde](#) [numerical-modelling](#) [physics-simulation-library](#) [computational-physics](#) [c-plus-plus](#) [python](#) [Manage topics](#)

**5,614** commits **100** branches **10** releases **30** contributors

Branch: [next](#) [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

This branch is 1967 commits ahead, 2 commits behind master. [Pull request](#) [Compare](#)

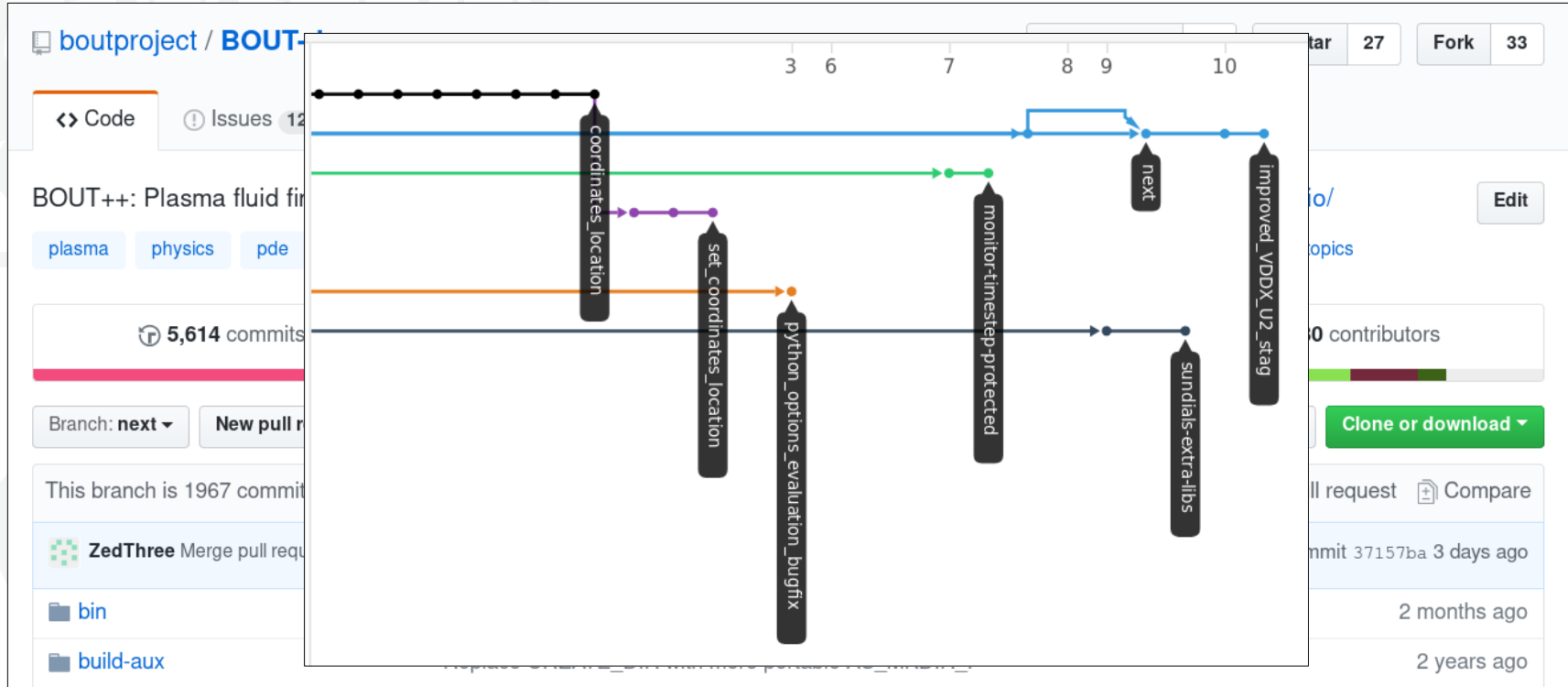
**ZedThree** Merge pull request [#1188](#) from boutproject/python-option-expressions-fix Latest commit 37157ba 3 days ago

[bin](#) Merge pull request [#1050](#) from boutproject/petsc-tidy 2 months ago

[build-aux](#) Replace CREATE\_DIR with more portable AS\_MKDIR\_P 2 years ago

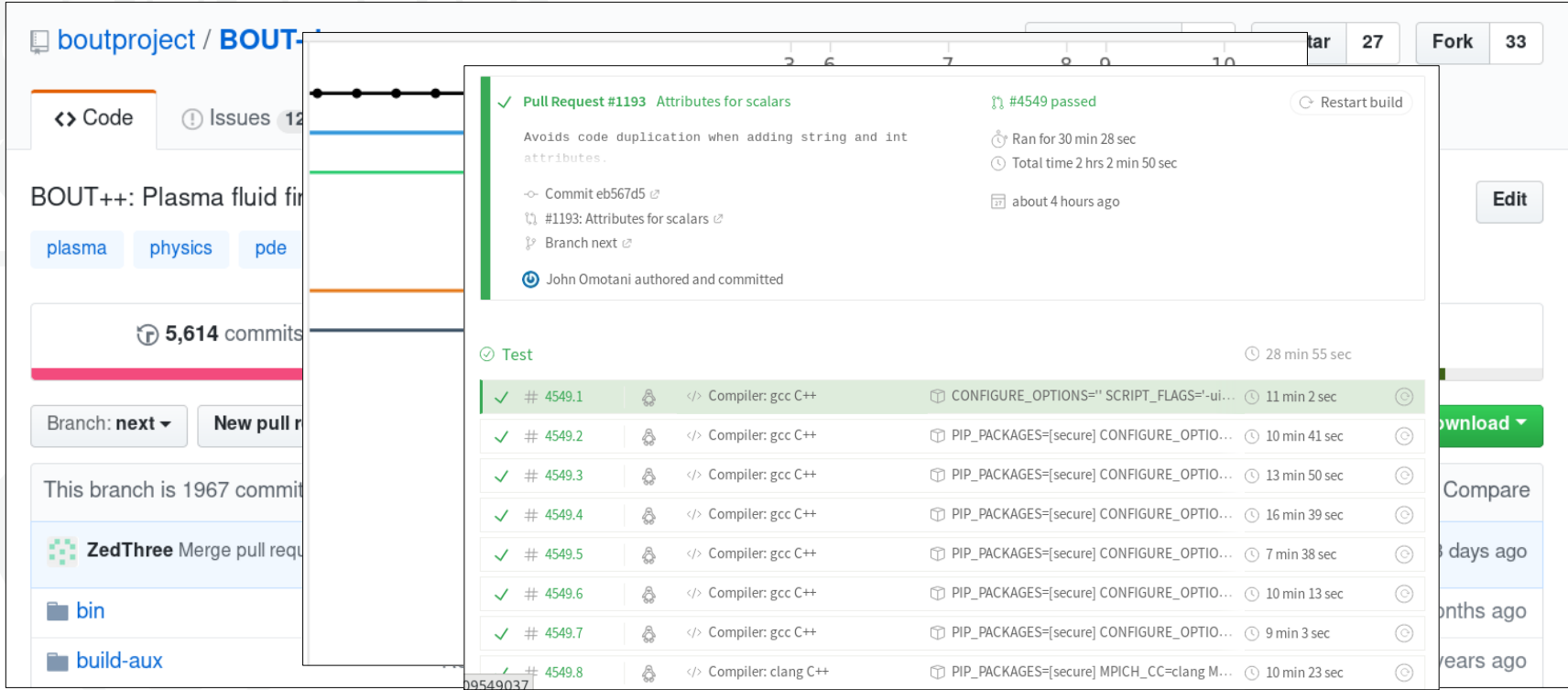
Code hosted on **Github**

# Development process



A system of branches with **master**, **next** and features

# Development process



The screenshot displays a GitHub pull request interface for the BOUT++ repository. The pull request, titled "Attributes for scalars", is in the "next" branch and has 4549 tests passed. The Travis CI build status is green, indicating successful automated testing. The interface includes a sidebar with repository navigation, a main area for the pull request details, and a table of test results.

**Repository:** boutproject / BOUT++

**Branch:** next

**Commits:** 5,614

**Tests:** 4549 passed

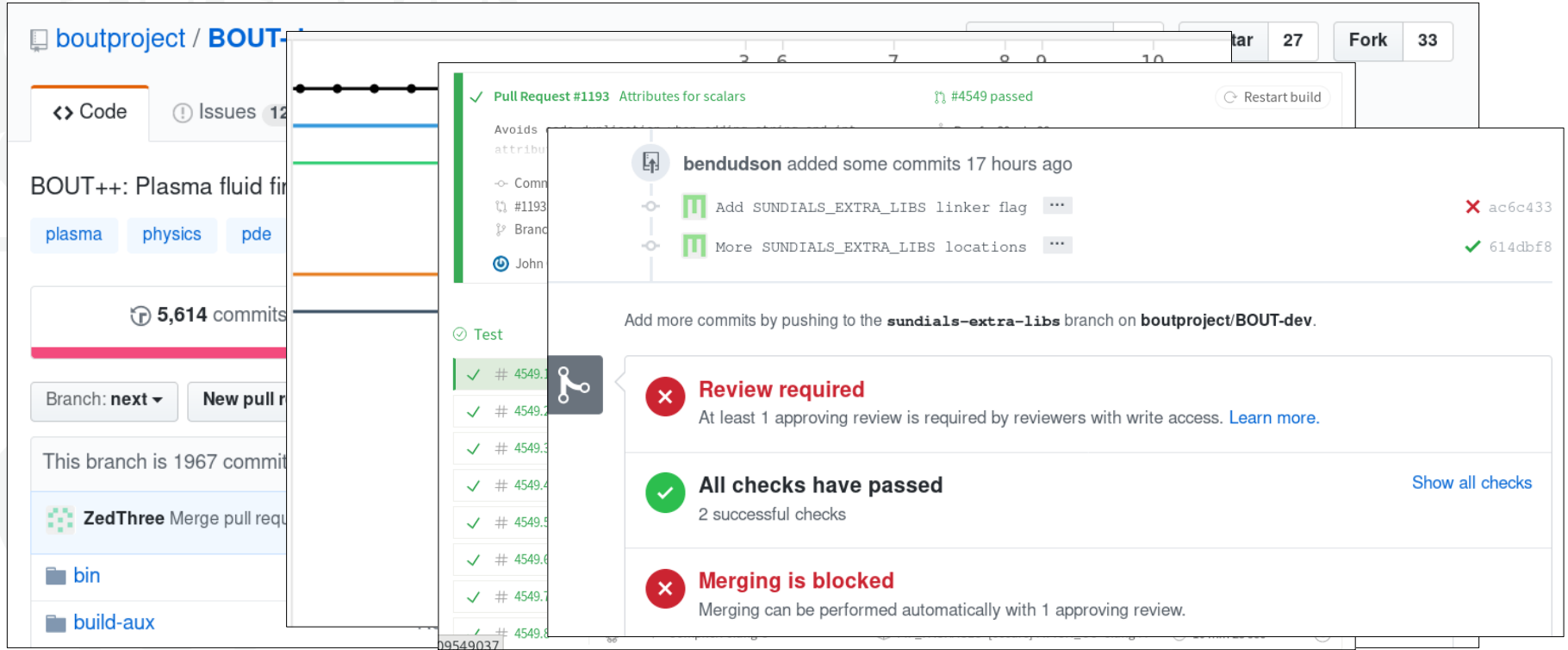
**Build Status:** Ran for 30 min 28 sec, Total time 2 hrs 2 min 50 sec

**Test Results:**

Test ID	Compiler	Configuration	Duration
# 4549.1	gcc C++	CONFIGURE_OPTIONS="-i...	11 min 2 sec
# 4549.2	gcc C++	PIP_PACKAGES=[secure] CONFIGURE_OPTIO...	10 min 41 sec
# 4549.3	gcc C++	PIP_PACKAGES=[secure] CONFIGURE_OPTIO...	13 min 50 sec
# 4549.4	gcc C++	PIP_PACKAGES=[secure] CONFIGURE_OPTIO...	16 min 39 sec
# 4549.5	gcc C++	PIP_PACKAGES=[secure] CONFIGURE_OPTIO...	7 min 38 sec
# 4549.6	gcc C++	PIP_PACKAGES=[secure] CONFIGURE_OPTIO...	10 min 13 sec
# 4549.7	gcc C++	PIP_PACKAGES=[secure] CONFIGURE_OPTIO...	9 min 3 sec
# 4549.8	clang C++	PIP_PACKAGES=[secure] MPICH_CC=clang M...	10 min 23 sec

Automated testing using Travis (unit, integrated and MMS tests) 39 / 42

# Development process

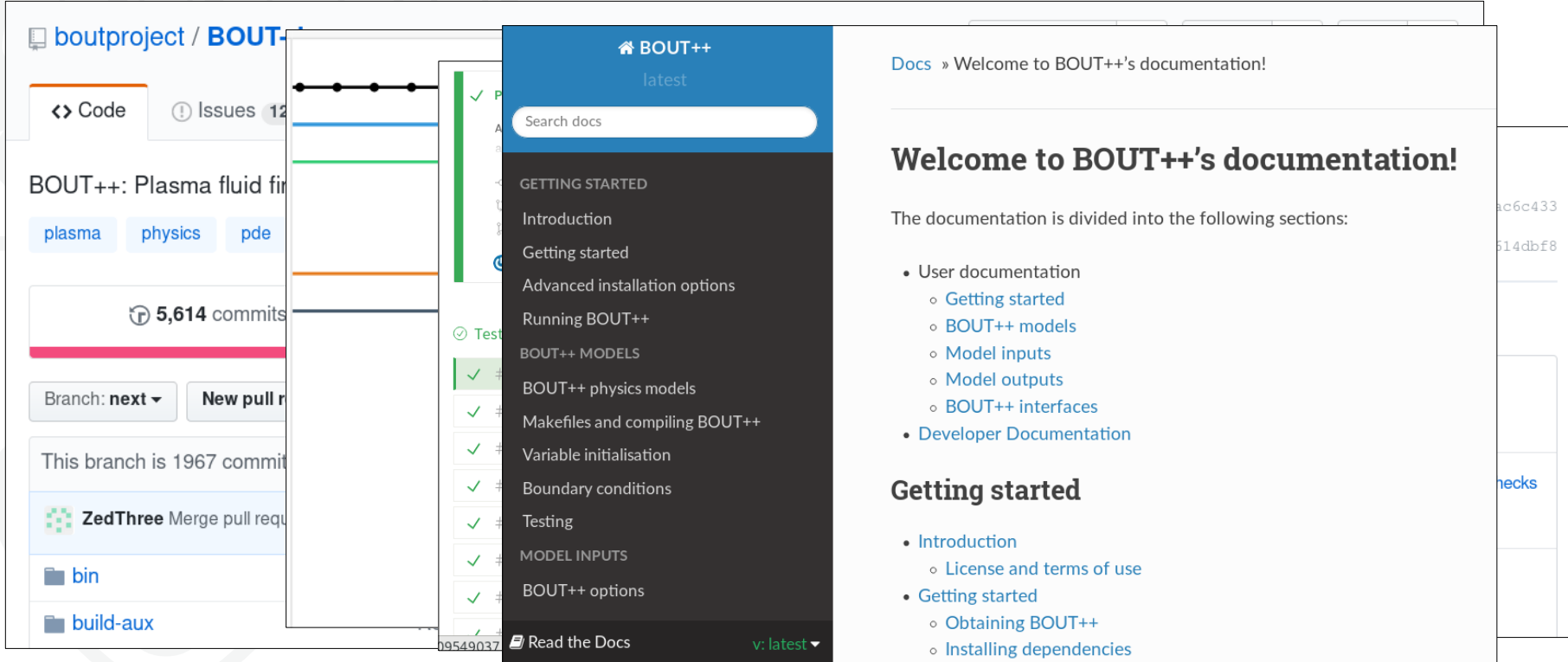


The screenshot displays a GitHub pull request for the BOUT++ project. The interface includes a sidebar on the left with navigation options like 'Code', 'Issues', and 'Pull requests'. The main content area shows the pull request details, including the title 'Pull Request #1193: Attributes for scalars', the author 'bendudson', and the status '17 hours ago'. A list of commits is visible, with the most recent one being '#4549.3'. The pull request is currently in the 'Review required' state, indicated by a red 'X' icon and the text 'Review required'. Below this, a green checkmark indicates 'All checks have passed' (2 successful checks). A red 'X' icon also indicates 'Merging is blocked' because 'At least 1 approving review is required by reviewers with write access.' The interface also shows a 'Restart build' button and a 'Fork' button.

Merges into next must pass tests and review



# Development process



The screenshot illustrates the development process for BOUT++, showing a GitHub repository, a Read the Docs page, and a table of contents.

**GitHub Repository (Left):** The repository is named `boutproject / BOUT++`. It shows the `Code` tab, a commit history of 5,614 commits, and a branch named `next`. A pull request from `ZedThree` is visible.

**Read the Docs Page (Right):** The page is titled `Welcome to BOUT++'s documentation!`. It lists the following sections:

- User documentation
  - Getting started
  - BOUT++ models
  - Model inputs
  - Model outputs
  - BOUT++ interfaces
- Developer Documentation

**Table of Contents (Center):** The table of contents is organized into sections:

- GETTING STARTED
  - Introduction
  - Getting started
  - Advanced installation options
  - Running BOUT++
- BOUT++ MODELS
  - BOUT++ physics models
  - Makefiles and compiling BOUT++
  - Variable initialisation
  - Boundary conditions
  - Testing
- MODEL INPUTS
  - BOUT++ options

The table of contents also includes a `Read the Docs` link and a `v: latest` dropdown menu.

Documentation automatically pulled from Github to Read the Docs

# Summary

- Fusion plasma physics has many interesting challenges
- BOUT++ has developed as a flexible tool to address a wide range of different problems.
- A good community of users and developers are pushing the code in new directions and adding capabilities
- Development is increasingly professional, with increasing emphasis on maintainability, reproducibility and correctness.

We welcome new contributors and collaborators!